ESAIM: PROCEEDINGS AND SURVEYS, **October 2015, Vol. 51, p. 246-271**
A. Garivier et al, Editors

ONLINE LEARNING AND GAME THEORY. A QUICK OVERVIEW WITH RECENT RESULTS AND APPLICATIONS [*]

Mathieu Faure[1], Pierre Gaillard[2], Bruno Gaujal[3] and Vianney Perchet[4]

Abstract. We study one of the main concept of online learning and sequential decision problem known as regret minimization. We investigate three different frameworks, whether data are generated accordingly to some i.i.d. process, or when no assumption whatsoever are made on their generation and, finally, when they are the consequences of some sequential interactions between players.

The overall objective is to provide a comprehensive introduction to this domain. In each of these main setups, we define and analyze classical algorithms and we analyze their performances. Finally, we also show that some concepts of equilibria that emerged in game theory are learnable by players using online learning schemes while some other concepts are not learnable.

Introduction

In sequential decision problems, or *online learning*, a decision maker takes sequentially decisions, for instance to classify some data, and his decisions (or *actions*) give him a sequence of rewards (or, alternatively, of losses). His objective is to maximize his cumulative rewards or, in an equivalent way, to minimize his *regret*. A possible interpretation of the latter is "the difference between the highest cumulative reward obtained by choosing at each stage the same action and the actual cumulative reward obtained". Stated otherwise, regret is the difference between what the decision maker could have got had he known in advance the empirical distribution of outcomes (the rewards associated to each action) and what he actually got, hence the name of "regret".

This concept has been introduced by Hannan [1957] and have got a lot of interest recently (see the fundamental textbook Cesa-Bianchi and Lugosi [2006] and references therein). One of the main examples of applications concerns websites that display advertisements each time a new visitor arrives. A website gets paid if the user clicks on the banner displayed. As a consequence, one must learn which ad has the highest probability of clicks, as displaying it generates the highest revenue. Moreover, the learning should be as fast as possible: displaying too many ads without clicks might give better estimates of low probabilities, but the original problem is not really the estimation of these probabilities than finding the optimal one.

Those problems are usually called "bandit" problems, in reference to the slot machines that can be found in a casino (see the recent survey Bubeck and Cesa-Bianchi [2012]). When entering a casino, a decision maker faces a large number of machines (hence the name of "multi-armed bandit") and the expected reward of each

---

[*] *The last author thanks the French Agence Nationale de la recherche under grants Jeudy ANR-10-BLAN–0112 and Gaga ANR-13-JS01-0004*

[1] Aix-Marseille University, CNRS and EHESS, Centre de la vieille Charité, 2 rue de la Charité, 13002 Marseille, France

[2] EDF R&D, Clamart, France & GREGHEC (HEC Paris, CNRS), Jouy-en-Josas, France

[3] INRIA Grenoble Rhône-Alpes, LIG laboratoire d'Informatique de Grenoble, MESCAL

[4] Laboratoire de Probabiliés et Modèles Aléatoires, Université Paris Diderot, Paris, France & Equipe Sierra, INRIA - ENS Paris, Paris, France

machine might differ. In order to maximize his gains (or, to be more realistic, to minimize his losses), he pulls arms of machines in some sequential order with the objective of finding the best machine as fast as possible.

These sequential decision problems simultaneously combine estimation and optimization. As a matter of fact, the main difficulty is deciding how much time/effort/money should be spent on estimation, to get precise and/or accurate prediction, and on optimization, i.e., to use the information already gathered to obtain the highest reward, maybe without getting additional information.

We shall consider three main frameworks.

- In the first one, we will assume that the rewards obtained by a decision (or an arm) are i.i.d. and independent from each other, which is typically the case in the casino example. In that case, we shall show that the regret can be uniformly bounded if the rewards of all machines are observed at each time (or with additional a-priori knowledge on the distribution of rewards) and increases logarithmically otherwise, see Lai and Robbins [1985].
- In the second more general setup, we shall remove this stochasticity assumption by assuming that the sequence of rewards can be arbitrary; even more generally, rewards might be adaptive to the past choices of the decision maker. This framework can be seen as a game between the decision maker and Nature that chooses at each stage all the possible rewards. A key property is that no assumptions are made on the behavior, or the objectives of Nature: a strategy of the decision maker must be good against any sequence of rewards. Quite surprisingly, and as shown in the second section, it is possible to devise strategies with a sub-linearly increase of regret, typically in $\sqrt{n}$, where n is the number of stages, see Auer et al. [2002b]. This frameworks is usually referred to as "adversarial"' (or individual sequences).
- In the last framework, we may assume that rewards are actually generated by other players that might also be minimizing their regret. The main question is whether the behavior of the player converges, in a sense to be defined properly, to some solution concept of game theory. This is the central question of the last section, see also Hart and Mas-Colell [2000].

As the objective is to provide a comprehensive introduction to this domain, the detailed organization of the paper is the following.

In Section 1, we consider learning in stochastic environment. The general model is detailed in Subsection 1.1, then we study the full monitoring case (that can be used as a benchmark to evaluate complexity of other models) in Subsection 1.2, the more general bandit monitoring framework in Subsection 1.3 and then we compare whether guarantees of the full monitoring case can be achieved with partial monitoring in Subsection 1.4. We conclude this section with some possible extensions, detailed in Subsection 1.5.

The adversarial case is studied in Section 2.The model is developed in Subsection 2.1 and a well known and used algorithm based on exponential weights is also described there. In the next Subsection 2.2, we provide a generic and useful tool, based on "stochastic approximations", to obtain asymptotic performance of some class of strategies. We then show in Subsection 2.3 that, in this framework, smaller regret bounds can sometimes be achieved under some regularity assumptions upon the sequences of data generated by Nature. We also introduce more general versions of regret in Subsection 2.4 and we show how to minimize them. This section is concluded in Subsection 2.5 with a detailed application of this theory.

We consider in the last Section 3 the game theoretic framework, whose general model is recalled in Subsection 3.1. We first introduce the concept of learnable equilibria and we show that the classical "Nash equilibria" are typically not learnable in Subsection 3.2. In Subsection 3.3, we consider learning equilibria in a specific subclass of games, potential games. In the final Subsection 3.4, learning in a distributed way is considered and studied.

1. Learning in Stochastic Environment

We first consider the "stochastic environment" with i.i.d. data obtained and treated sequentially.

1.1. **Model, Estimation Procedures and Regret**

We consider the sequential decision problems where an agent (or a decision maker, an algorithm, a player, depending on the context) takes at each stage $n \in \mathbb{N}$ a decision k_n in a finite set $\mathcal{K} := \{1, \dots, K\}$. In the multi-armed bandit langage, he "pulls the arm k_n". The reward received depends both on the state $n \in \mathbb{N}$ and on the decision $k_n \in \mathcal{K}$. It is denoted by $X_n^{(k_n)} \in [0,1]$ where $\{X_n^{(k)}\}_{n \in \mathbb{N}}$ are K i.i.d. processes in $[0,1]$ whose distributions and expectations $\mu^{(k)} := \mathbb{E}[X_n^{(k)}]$ are unknown.

The decision k_n can be a (random) function of the information available to the agent up to time n. We distinguish two main scenarios:

- Either k_n can depend on all the previous values of all processes, i.e., it can depend on $\{X_m^{(k)}, k_m;\, m \in \{1, \dots, n-1\},\, k \in \mathcal{K}\}$ – this framework is referred to as *with full monitoring* –,
- or k_n can only depend on the sequence of rewards of the agent, i.e., on $\{X_m^{(k_m)}, k_m;\, m \in \{1, \dots, n-1\}\}$ – a framework referred to as *with bandit monitoring*.

Stated otherwise, with bandit monitoring, the agent observes only the realization of the chosen process (the arm pulled) while he observes all the realizations (even of the processes he did not choose, i.e., the arm he did not pull) with full monitoring. We call a strategy (or algorithm, policy, decision rule, etc.) the mapping associating to any finite history the decision to be taken.

The agent aims at maximizing the cumulative expected reward $\sum_{m=1}^n \mu^{(k_m)} = \mathbb{E} \sum_{m=1}^n X_m^{(k_m)}$. Depending on the value of the parameters $\mu^{(1)}, \dots, \mu^{(K)}$, this quantity can be arbitrarily large, i.e., of the order of n, or arbitrarily small, i.e., of the order of 1. As a consequence, the overall objective is usually rewritten in terms of minimization of *regret* R_n defined by (all the following definitions are equivalent):

$$R_n := \max_{k \in \mathcal{K}} n\mu^{(k)} - \sum_{m=1}^n \mu^{(k_m)} = n\mu^\star - \sum_{m=1}^n \mu^{(k_m)} = \sum_{m=1}^n \Delta^{(k_m)} = \sum_{k \in \mathcal{K}} \Delta^{(k)} N_n[k],$$

where

$$\mu^\star = \max_{k \in \mathcal{K}} \mu^{(k)}, \quad \Delta^{(k)} := \mu^\star - \mu^{(k)}, \quad N_n[k] = \sharp\{m \leq n;\, k_m = k\}.$$

The parameter $\Delta^{(k)}$ is referred to as the *gap* of the arm k and $N_n[k]$ is the number of times arm k was pulled before stage n. We emphasize here that the regret R_n is a random variable since it is defined through the sequence $\{k_n\}$ which depends on the random sequences $\{X_n^{(k)}\}$. We aim at devising strategies guaranteeing that the expected regret $\mathbb{E}[R_n]$ grows sub-linearly, logarithmically or even that it is uniformly bounded.

With bandit monitoring, the difficulty of this problem is that it combines the estimation of the $\mu^{(k)}$ and the optimization of the rewards. The first one is usually called the *exploration* and the second one the *exploitation*. We shall conclude this section by showing that the exploration, i.e., the mere estimation of the means $\mu^{(k)}$, is by itself rather easy.

Indeed, given an i.i.d. process $Z_n \in [0,1]$, an unbiased estimator of its mean μ is simply the empirical average $\overline{Z}_n := \sum_{m=1}^n Z_m/n$ which concentrates exponentially fast around μ, thanks to Hoeffding's inequality (see, e.g., Boucheron et al. [2013] for a recent textbook on concentration inequalities):

$$\mathbb{P}\Big\{\overline{Z}_n - \mu \geq \varepsilon\Big\} \leq e^{-2n\varepsilon^2}, \quad \forall \varepsilon \geq 0. \tag{1}$$

This simple exponential inequality has a crucial consequence: estimating the mean μ through the empirical averages $\overline{Z}_n$ makes only a *finite* number of mistakes bigger than ε in expectation, i.e., the number of stages where $\overline{Z}_n$ is ε-away from μ is uniformly bounded. Indeed, for all $\varepsilon > 0$,

$$\mathbb{E}\Big[\sum_{m \in \mathbb{N}_*} \mathbb{1}\big\{|\overline{Z}_n - \mu| \geq \varepsilon\big\}\Big] \leq 2 \sum_{m \in \mathbb{N}_*} e^{-2n\varepsilon^2} = \frac{2}{e^{2\varepsilon^2} - 1} \leq \frac{1}{\varepsilon^2}\,.$$

This bound can be slightly improved to $\log(2)/\varepsilon^2$ if $2\varepsilon^2 \leq \log(2)$ or using more precise concentration inequalities.

1.2. The benchmark: Full Monitoring

We first consider the easiest problem, with full monitoring. We recall that, in this setup, the decision $k_n \in \mathcal{K}$ can depend on all the precedent values $X_m^{(k)}$, for all $m \leq n-1$ and all $k \in \mathcal{K}$. This simple framework shall be used as a "benchmark" to evaluate the difficulty of the bandit monitoring as well as the performance of algorithms.

In this framework, one can devise a very simple strategy with impressive guarantees.

Optimal Strategy with Full Monitoring

Choose $k_{n+1} \in \arg\max_{k\in\mathcal{K}} \overline{X}_n^{(k)}$

Let $k \in \mathcal{K}$ be such that $\Delta^{(k)} = \mu^\star - \mu^{(k)} > 0$, then it holds that $\overline{X}_n^\star - \overline{X}_n^{(k)}$ is smaller than 0 on, at most, $1/\left(\Delta^{(k)}\right)^2$ stages, which is therefore a uniform bound on $\mathbb{E}[N_n[k]]$. As a consequence, we directly obtain that the expected regret of this strategy is bounded as

$$\mathbb{E}[R_n] \leq \sum_{k:\Delta^{(k)}>0} \frac{1}{\Delta^{(k)}} \,. \tag{2}$$

Actually, the upper-bound can be drastically improved, but the proofs are more involved. For instance, if all $\Delta^{(k)}$ are equal to some Δ, then it is possible to show that

$$\mathbb{E}[R_n] \lesssim \frac{\log(K)}{\Delta},$$

where the notation $\lesssim$ means that there exists some universal constant $c > 0$ (independent of all the parameters of the problem at hand) such that $\mathbb{E}[R_n] \leq c\log(K)/\Delta$

A particular interpretation of these results is that the expected regret is bounded uniformly in n. But there might exist trivial better upper-bounds for smaller values of n. For instance, if $n = 1$, $R_1 \leq 1 \ll \sum 1/\Delta^{(k)}$. This is a direct consequence of the fact that the main argument was that $\overline{X}_n^\star - \overline{X}_n^{(k)} \leq 0$ on, at most, $1/\left(\Delta^{(k)}\right)^2$ stages; obviously, when n is small enough, n is a better trivial upper bound on this number.

This idea can be used to obtain *worst case* (or distribution independent) bounds, i.e., guarantees on the expected regret $\mathbb{E}[R_n]$ that hold for all values of the gaps $\Delta^{(k)}$, but at a fixed horizon n. For instance, Equation (2) can be rephrased into

$$\mathbb{E}[R_n] \leq \sum_{k:\Delta^{(k)}>0} \frac{1}{\Delta^{(k)}} \wedge n_k\Delta^{(k)} \leq \sqrt{Kn},$$

where n_k is the expected number of stages where arm k is pulled before stage n. The last upper-bound comes from a simple optimization over all n_k and Δ_k, with the constraint that $\sum_k n_k = n$ and $\Delta^{(k)} \in [0,1]$. As before, this worst-case bound can be greatly improved; for instance, when all $\Delta^{(k)}$ are equal to some Δ, then

$$\mathbb{E}[R_n] \lesssim \frac{\log(K)}{\Delta} \wedge n\Delta = \sqrt{\log(K)n}\,.$$

We emphasize here the difference between *distribution-dependent* bounds (that depend on $\Delta^{(k)}$) and worst-case bounds. The former are given for fixed values of $\Delta^{(k)}$ and hold uniformly in $n \in \mathbb{N}$, while the latter are given for a fixed value of n and are true for any value of $\Delta^{(k)}$: for any n, they are obtained by choosing the worst possible gaps.

1.3. Performance Guarantees with Bandit Monitoring

We now consider the *bandit monitoring*, a more realistic framework for many applications in mind. We recall that the main constraint is that all the values $\{X_m^{(k)};\, m \leq n, k \in \mathcal{K}\}$ are not observed and the choice of k_{n+1} can only depends on $\{X_m^{(k_m)};\, m \leq n\}$.

In that framework, it is not possible, by definition, to choose $k_{n+1} \in \arg\max_{k\in\mathcal{K}} \overline{X}_n^{(k)}$, since those averages are not computable by the decision maker. The only computable averages are

$$\widehat{X}_n^{(k)} = \frac{\sum_{m=1}^n \mathbb{1}\{k_m = k\} X_m^{(k)}}{N_n[k]},$$

where we recall that $N_n[k]$ is the number of stages where arm k has been pulled before stage n. Without loss of generality, we can always assume that the K arms are each pull once during the first K stages, i.e., $k_m = m$ for $m \in \{1, \dots, K\}$, so that $\widehat{X}_n^{(k)}$ are well defined after K stages (otherwise, follow the convention $0/0 = 1$). The set of K first stages is called the *initialization phase*.

It is rather immediate to understand that choosing $k_{n+1} \in \arg\max \widehat{X}_n^{(k)}$ is doomed to incur a linear regret. Indeed, assume that $K = 2$, that $X_n^{(1)}$ is a Bernoulli of parameter $1/2$ and $X_n^{(2)} = 1/4$ deterministically. Then $\widehat{X}_2^{(1)} = 0$ with probability $1/2$ while $\widehat{X}_n^{(2)}$ is always equal to $1/4$. As a consequence, the aforementioned algorithm will always choose $k_n = 2$ which gives an expected regret of at least $n/8$.

The reason the regret grows linearly is that there are no theoretical guarantees that $\widehat{X}_n^{(k)}$ will converge to its expectation $\mu^{(k)}$, because $N_n[k]$ might not increase to infinity, as in the simple counter-example. An ingenuous way to tackle this issue is to define a *confidence interval* around each $\widehat{X}_n^{(k)}$ in which $\mu^{(k)}$ belongs with arbitrarily high probability. The strategy is then rather simple and consists in choosing the interval with the highest upper-bound since it might contain the highest mean.

This strategy is called UCB for *Upper Confidence Bound*, see Auer et al. [2002a], and is defined as follows.

```
Upper Confidence Bound (UCB)
    i) Initialization phase: For $n \leq K, k_n = n$
    ii) Choose $k_{n+1} \in \arg\max_{k\in\mathcal{K}} \widehat{X}_n^{(k)} + \sqrt{2\frac{\log(n)}{N_n[k]}}$
```

This strategy has the following guarantee:

$$\mathbb{E}[R_n] \lesssim \sum_{k:\Delta^{(k)}>0} \frac{\log(n)}{\Delta^{(k)}} + \Delta^{(k)}\ . \tag{3}$$

This result can be proved rather immediately without noise, i.e., when $X_n^{(k)} = \mu^{(k)}$ for all n and k. Indeed, a suboptimal arm k is played at stage n if $n = k$ (during the initialization phase) or if

$$\mu^\star + \sqrt{\frac{2\log(n)}{N_n[\star]}} \leq \mu^{(k)} + \sqrt{\frac{2\log(n)}{N_n[k]}} \implies N_n[k] \leq \frac{2\log(n)}{\left(\Delta^{(k)}\right)^2}.$$

The presence of noise is handled using Hoeffding's inequality that ensures that $\widehat{X}_n^{(k)}$ is equal to $\mu^{(k)}$ up to an error term of order $\sqrt{2\frac{\log(n)}{N_n[k]}}$ with a probability larger than $1 - 1/n^4$. As a consequence, the expected number of stages where $\mu^{(k)}$ is not in his confidence interval is uniformly bounded in expectation and on the other stages the same argument applies.

The algorithm UCB had a huge impact because it was the first one bounding the regret with a logarithmic, non-asymptotic dependency in n and for any distribution of rewards, as long as $X_n^{(k)}$ belongs to $[0,1]$. It even matches a lower bound since when $K=2$ and no matter the algorithm, there will always exist a set of distributions of $X_n^{(1)}$ and $X_n^{(2)}$ with $|\mu^{(1)}-\mu^{(2)}|=\Delta$ and such that $\limsup_n \mathbb{E}[R_n]/\log(n) \gtrsim 1/\Delta$, see Lai and Robbins [1985].

On the other hand, the performance of UCB is sub-optimal in the worst-case sense. Indeed, consider the case for simplicity where all the $\Delta^{(k)}$ are equal to the same Δ; in that case, Equation (3) can be rewritten into

$$\mathbb{E}[R_n] \lesssim \frac{K\log(n)}{\Delta} \wedge n\Delta$$

and the right-hand side is maximized, when n is fixed, for $\Delta = \sqrt{K\log(n)/n}$ which gives a regret bounded as $\sqrt{K\log(n)n}$. The dependency in n is suboptimal as it is known that regret can always be bounded in $\sqrt{Kn}$, see Vovk [1990].

In order to remove this extra logarithmic term in the worst case, the UCB algorithm has been adapted in different ways; the common idea is to modify the confidence term. When the horizon of the interaction $N \in \mathbb{N}$ if fixed and known in advance, the MOSS algorithm (where MOSS stands for Minimax Optimal Strategy in the Stochastic case, see Audibert and Bubeck [2010]) turns the confidence term into

$$\sqrt{\frac{\log\left(\frac{N}{KN_n[k]}\right)}{N_n[k]}} \quad \text{if} \quad N_n[k] \le \frac{N}{K} \quad \text{and} \quad 0 \quad \text{otherwise.}$$

This algorithm guarantees that, with the notation $\Delta_{\min} = \min\{\Delta^{(k)};\Delta^{(k)}>0\}$:

$$\mathbb{E}[R_N] \lesssim \frac{K}{\Delta_{\min}} \overline{\log}\left(\frac{N\Delta_{\min}^2}{K}\right), \quad \text{where} \quad \overline{\log}(\cdot) = \max\{1, \log(\cdot)\}.$$

More importantly, it also ensures that $\mathbb{E}[R_n] \lesssim \sqrt{KN}$ in the worst case analysis.

Another idea is to draw arms alternatively until one is proved to be, with very high probability, better than another one, see Perchet and Rigollet [2013]. When it happens, the supposed to be suboptimal arm is stop being pulled. The testing criterion is rather simple as an arm k is declared better than another arm k' at some stage n if

$$\widehat{X}_n^{(k)} - \sqrt{\frac{\log\left(\frac{N}{N_n[k]}\right)}{N_n[k]}} \gtrsim \widehat{X}_n^{(k')} + \sqrt{\frac{\log\left(\frac{N}{N_n[k']}\right)}{N_n[k']}} \ .$$

This algorithm is called SE for Successive Elimination, it also requires the knowledge of the horizon N and it has the following guarantee

$$\mathbb{E}[R_N] \lesssim \sum_{k:\Delta^{(k)}>0} \frac{\overline{\log}\left(N(\Delta^{(k)})^2\right)}{\Delta^{(k)}} \quad \text{and} \quad \mathbb{E}[R_N] \lesssim \sqrt{K\log(K)N} \ .$$

Since $K/\Delta_{\min}$ and $\sum_k 1/\Delta^{(k)}$ are not comparable, the two algorithms MOSS and SE are not comparable either and their relative performances depend on the set of parameters $\{\Delta^{(k)}\}$. In the worst case analysis, the SE algorithm suffers an extra $\sqrt{\log(K)}$ term but, in any case, it is smaller than the $\sqrt{\log(n)}$ that appears with UCB. On the other hand, the SE algorithm is simpler in the sense that remaining arms are pulled one after the other, in a more predictable way than when following UCB and MOSS.

One of the main drawbacks of both algorithms is that the horizon N must be known in advance. A variant of UCB, called UCB-2 (its analysis is more involved than the original version of UCB, Auer et al. [2002a], and it is also based on a modification of the confidence term), actually achieves the same guarantee as SE at all stages $n \in \mathbb{N}$ (and not just at a fixed horizon).

1.4. Matching the Performances of Full Monitoring

A natural question that arises is whether the performances attainable with full monitoring can be attained in the bandit setting, even when $K = 2$. Without additional assumptions (see also Section 2.3), the answer is simply no. Indeed, assume that there are two possible worlds (and that the decision maker knows that) such that $X_n^{(1)} = 1/2$ at all stages and $X_n^{(2)}$ is drawn accordingly to a $\mathcal{N}(1/2 + \Delta, 1)$ in the world 1 and accordingly to $\mathcal{N}(1/2 - \Delta, 1)$ in the world 2. Then any algorithm will suffer a regret of the order of $\log(n\Delta^2)/\Delta$ in one of the two worlds (the fact that the support of $X_n^{(2)}$ is not bounded is irrelevant, as all the previous results hold for sub-Gaussian variables), see Bubeck et al. [2013]. As a consequence, knowing the gap between the optimal arm and the sub-optimal arm is not sufficient to obtain a uniformly bounded regret in $1/\Delta$ as with full monitoring.

On the other hand, if $\mu^\star$ and Δ are both known then regret can be uniformly bounded as follows (see Bubeck et al. [2013] or Lai and Robbins [1984] for an asymptotic version). At all stages such that at least one of $\widehat{X}_n^{(1)}$ or $\widehat{X}_n^{(2)}$ is above the threshold $\mu^\star - \Delta/2$, play $k_n \in \arg\max \widehat{X}_n^{(k)}$. If both of them are below the threshold, then play each arm once in the following two stages. Equation (1) implies that $\widehat{X}_n^{(2)}$ is going to be above the threshold and $\widehat{X}_n^{(1)}$ below it only $2/\Delta^2$ times. As a consequence, the suboptimal arm is going to be played at most $4/\Delta^2$ times. This gives an expected regret uniformly bounded in $1/\Delta$.

If there are more than 2 arms, and all $\widehat{X}_n^{(k)}$ are below the threshold, then arm k should be chosen at random, with a probability that decreases with the distance to the threshold. Indeed, one can show that regret is then smaller than

$$\mathbb{E}R_n \lesssim \sum_{k:\Delta^{(k)}>0} \frac{\log(\Delta^{(k)}/\Delta_{\min})}{\Delta^{(k)}}.$$

There are other possible ways to obtain bounded regret, with other a-priori knowledges on the set of parameters $\{\Delta^{(k)}\}$, such as the knowledge of only $\mu^\star$, of an intermediate value between $\mu^\star$ and $\mu^\star - \Delta$, etc.

1.5. Extensions

There are many possible extensions of this framework. Among them, we can mention

Continuous bandits: One of the key feature in multi-armed bandit was that the number of arms was finite, equal to K. In continuous bandits, arms may be elements of some compact, non-necessarily finite, set. For instance, an arm can be any point of the interval $[0, 1]$, or more generally the hypercube $[0, 1]^d \subset \mathbb{R}^d$. In this case, to any point $x \in [0, 1]^d$, we associate an i.i.d. process $X_n^{(x)} \in [0, 1]$ whose expectation is $\mu(x)$.

Without any regularity assumption on $\mu(\cdot)$ the regret can decrease arbitrarily slowly. We might assume for instance that $\mu(\cdot)$ is 1-Lipschitz, β-Hölder (i.e., $|\mu(x) - \mu(y)| \leq \|x - y\|^\beta$), etc. And under such regularity assumptions, regret is sublinear.

An algorithm is rather simple, Kleinberg [2004], and we may assume for simplicity that the horizon n is known in advance. Let $\mathcal{K} := \{x^{(1)}, \ldots, x^{(K)}\}$ be an ε discretization of $[0, 1]^d$ and run the algorithm

MOSS on this ε discretization. Then one immediately obtains that

$$\begin{aligned}\mathbb{E}R_N &= N \max_{x\in[0,1]^d} \mu(x) - \sum_{m=1}^{N} \mu(x_m)\\ &\le N\left(\max_{x\in[0,1]^d} \mu(x) - \max_{x^{(k)}\in\mathcal{K}} \mu(x^{(k)})\right) + \left(\max_{x^{(k)}\in\mathcal{K}} \mu(x^{(k)}) - \sum_{m=1}^{N} \mu(x_m)\right)\\ &\lesssim N\varepsilon^\beta + \sqrt{KN} \lesssim N\varepsilon^\beta + \sqrt{\frac{1}{\varepsilon^d} N} \lesssim N^{\frac{\beta+d}{2\beta+d}},\end{aligned}$$

if $\varepsilon = N^{-1/(2\beta+d)}$ because K, the size of the discretization, is of the order of $1/\varepsilon^d$. This kind of results can be further improved with additional assumptions on $\mu(\cdot)$, for instance on its local behavior near the optimum as in Bubeck et al. [2011].

Linear bandits: Linear bandits are special cases of continuous bandits where the function $\mu(\cdot)$ is linear. In particular; $\mu(\cdot)$ is Lipschitz and so one might expect an upper-bound in $n^{(1+d)/(2+d)}$. Actually, it is possible to get a drastic improvement in this rate of convergence as there exist algorithms bounding regret in poly$(d)\sqrt{n}$, thus the rate of convergence is independent of the dimension, as in finite multi-armed bandit. Getting the right dependency in d in the polynomial is a major challenge, tackled for instance in Abernethy et al. [2008], Dani et al. [2008].

Bandits with covariates: The framework of bandits with covariates, see Woodroofe [1979], is described by two sets. A finite set of arms $\mathcal{K} = \{1, \dots, K\}$ and a convex compact set of *covariates* $\mathcal{Z} = [0,1]^d$. At each stage $n \in \mathbb{N}$ and before choosing k_n, a covariate $Z_n \in \mathcal{Z}$ is drawn accordingly to some distribution, say uniformly on $\mathcal{Z}$. Choosing arm k then yields a reward $X_n^{(k)}$ of expectation $\mu^{(k)}(Z_n)$ where all the mapping $\mu^{(k)}$ are sufficiently regular, for instance β-Hölder, otherwise regret can decrease arbitrarily slowly.

In that framework, the optimal strategy chooses at stage $k^*[Z_n] \in \arg\max_k \mu^{(k)}(Z_n)$, so we define the optimal value mapping $\mu^\star(z) = \mu^{(k^*[z])}$. Notice that a strategy is described, at stage n, as a function that assigns to any possible value Z a choice $k_n(Z) \in \mathcal{K}$. As a consequence, expected regret is defined as

$$\mathbb{E}R_n = \sum_{m=1}^{n} \mathbb{E}\left[\mu^\star(Z) - \mu^{(k_n(Z))}(Z)\right]$$

Without additional assumptions on the separation between mappings $\mu^{(k)}(\cdot)$, which is the equivalent of gaps in the multi-armed bandits case, regret can be bounded as

$$\mathbb{E}R_n \le n\left(\frac{K\log(K)}{n}\right)^{\frac{\beta}{2\beta+D}},$$

which has the same dependency in n has β-Hölder continuous bandit. The previous bound must be understood as a slow, worst-case type of bound. It is possible to obtain fast distribution dependent bounds by introducing a concept generalizing the gaps of multi-armed bandit, based on the margin condition. We say that the distribution of $X_n^{(k)}, Z_n$ satisfies the α-margin condition if

$$\mathbb{P}\Big\{0 < \mu^\star(Z) - \mu^\sharp(Z) \le \delta\Big\} \le c\delta^\alpha, \quad \forall\delta > 0$$

where $\mu^\sharp(Z)$ is the second best strategy defined by $\mu^\sharp(Z) = \max_k \left\{\mu^{(k)}; \mu^{(k)} < \mu^\star\right\}$ whenever it is defined, otherwise $\mu^\sharp(Z) = \mu^\star(Z)$. This margin condition represents how *separated* the optimal value and the second optimal value are: a small value of α indicates that the problem is difficult, similarly to a small value of Δ in multi-armed bandits, while problems with a large value of α are easier. We emphasize here that the regularity of $\mu^{(k)}$, i.e., the coefficient β is known, while α is unknown.

An adapted version of Successive Elimination, developed in Perchet and Rigollet [2013], that discards arms on a partition of $[0,1]^d$ refined over time achieves the following guarantee:

$$\mathbb{E}R_n \leq n \left(\frac{K \log(K)}{n} \right)^{\frac{\beta(1+\alpha)}{2\beta+D}} .$$

The dependency in n, and most probability not the one in K (because of the logarithmic term $\log(K)$), is optimal.

Structured bandits: There are many different examples of structured bandits, as *sparse bandits*. The vector $(\mu^{(1)}, \dots, \mu^{(k)})$ is s-sparse if all but s coordinates are non equal to zero. In order to get non-trivial results, we may assume in this case that $X_n^{(k)}$ belongs to $[-1,1]$ and not $[0,1]$. The typical idea would be to replace all dependencies in K by s, similarly to Gerchinovitz [2013].

In *combinatorial bandits*, we assume some structure on the set of actions. For instance, assume that $\mathcal{K} = \{0,1\}^d$ and choosing $k = (k_1, \dots, k_d) \in \mathcal{K}$ gives a payoff of $X_n^{(k)} := \sum_{i=1}^d k_i Z_n^{(i)}$ where $Z_n^{(i)}$ are i.i.d. random variables. We can even distinguish two types of bandit monitoring, whether only $X_n^{(k_n)}$ is observed or the values $\{k_i Z_n^{(i)}\}$. With full monitoring case, all the values $\{Z_n^{(i)}\}$ are observed, see Audibert et al. [2014].

Partial Monitoring: The bandit monitoring can be seen as a specific case of *partial monitoring*. It is described by an additional matrix H of size $K \times K$ whose components are elements of some Euclidean space $\mathbb{R}^d$. The observation after stage n is the vector $e_{k_n}^T H X_n$ where $X_n = (X_n^{(1)}, \dots, X_n^{(k)})$ and e_k is the unit canonical vector.

For instance, the bandit monitoring corresponds to $H = \mathrm{Id}$ with $d = 1$, while the full monitoring corresponds to $H_{k,\ell} = e_\ell$, with $d = K$. In the general case, it is possible that the regret cannot be guaranteed to have a sub-linear growth, for instance when $H = 0$. In that case, it is however possible to define and minimize a weaker notion of regret, see Lugosi et al. [2008], Perchet [2011], Rustichini [1999].

2. Non-Stochastic (or Adversarial) Environment

In this section, we remove the stochasticity assumption on the data. This general framework is motivated by, among others, the spam detection problem. Indeed, spam senders adapt their strategy to the new filters used and, as a consequence, the spam received might not be considered to be i.i.d.

2.1. Model and the Exponential Weight Algorithm

In this section, the stochasticity assumption on the processes $X_n^{(k)}$ is removed, i.e., they are not necessarily i.i.d. The value of $X_n^{(k)}$ might even depend on all the past values $\{X_m^{(k)},\ m \leq n-1,\ k \in \mathcal{K}\}$ and on $\{k_1, \dots, k_{n-1}\}$. This framework can be seen as *a game* between the decision maker and Nature: at each stage, depending on the past history, they choose "simultaneously[1]" $k_n \in \mathcal{K}$ and $X_n^{(k)}$. In this more general framework, we shall only focus on the full monitoring case, even though the bandit monitoring has also been widely studied, see for instance the survey Bubeck and Cesa-Bianchi [2012].

The notion of regret is similar to the one introduced before, up to the fact that $\mu^{(k)}$ are not defined:

$$R_n := \max_{k \in \mathcal{K}} \sum_{m=1}^n X_m^{(k)} - \sum_{m=1}^n X_m^{(k_m)} .$$

---

[1] The word simultaneously is merely used to precise that the values of $X_n^{(k)}$ are independent of k_n.

It is rather immediate to see that any deterministic strategy might have a linear regret, so the decision maker, in order to have a sub-linear regret, must choose actions at random. We denote by $p_n \in \Delta(\mathcal{K})$ the probability distribution over $\mathcal{K}$ accordingly to which $k_n \in \mathcal{K}$ is drawn, where $\Delta(\mathcal{K})$ stands for the set of probability distributions over the finite set $\mathcal{K}$. We emphasize that $p_n \in \Delta(\mathcal{K})$ depends on the past history. Using concentration inequalities for sums of martingale differences, it is possible to show that $\sum_{m=1}^n X_n^{(k_n)}$ is $\sqrt{n}$-close of $\sum_{m=1}^n p_n X_n$, where $p_n X_n$ stands for the expected value of $X_n^{(k_n)}$ given the past history. As a consequence, regret could be rewritten into

$$R'_n := \max_{p\in\Delta(\mathcal{K})} \sum_{m=1}^n pX_m - \sum_{m=1}^n p_m X_m = \max_{k\in\mathcal{K}} \sum_{m=1}^n X_m^{(k)} - \sum_{m=1}^n p_m X_m \ .$$

One of the most classical algorithm, called *exponential weight algorithm*, introduced in this setup in Littlestone and Warmuth [1994], Vovk [1990], is defined as follows.

Exponential Weight Algorithm, $\eta > 0$ some parameter.

- i) p_1 is the uniform distribution over $\mathcal{K}$
- ii) Choose $p_n^{(k)} = \dfrac{e^{\eta \sum_{m=1}^{n-1} X_m^{(k)}}}{\sum_{\ell=1}^K e^{\eta \sum_{m=1}^{n-1} X_m^{(\ell)}}}$

If the horizon N is known, the choice of $\eta = \sqrt{8\log(K)/N}$ ensures that

$$R'_N \leq \sqrt{N\log(K)/2}.$$

If the horizon is unknown, then choosing a varying parameter $\eta_n = \sqrt{8\log(K)/n}$ ensures that $R'_n \leq 2\sqrt{n\log(K)}$. The cost of not knowing the horizon lies therefore merely in the constants.

Since there are no assumptions on the sequences of rewards, the performances of this algorithm must be compared with the ones obtained in the worst case analysis in the stochastic setup. And in both cases, the guarantee in $\sqrt{n\log(K)}$ are identical. As a consequence, the exponential weight algorithm performs as well in a non-stochastic environment as the optimal algorithm in an i.i.d. environment.

Moreover, bounding regret in $\sqrt{n\log(K)}$ is also optimal in the minimax sense: no matter the algorithm, Nature can produce $X_n^{(k)}$ such that the regret is bigger than this quantity. Indeed, assume that $X_n^{(k)} \in \{0,1\}$ is chosen i.i.d. and $X_n^{(k)} = 0$ or $X_n^{(k)} = 1$ with probability 1/2. No matter the algorithm $\mathbb{E}p_n X_n = 1/2$ thus $\mathbb{E}\sum_{m=1}^n p_m X_m = n/2$. On the other hand, $\sum_{m=1}^n X_m^{(k)}$ follows approximatively a Gaussian $\mathcal{N}(n/2, n/4)$, as a consequence the maximum over k of these quantities has an expectation of the order of $n/2 + \sqrt{n\log(K)}$, and the regret is of the order of $\sqrt{n\log(K)}$, no matter the algorithm.

There are different ways to interpret the exponential weight algorithm (and each leads to a different proof of its performances)

- The first one is to see p_n as the gradient at the point $G_n := \left(\sum_{m=1}^n X_m^{(k)} - \sum_{m=1}^n p_m X_m\right)_{k\in\mathcal{K}} \in \mathbb{R}^K$ of the potential function Φ_η that associates to any $Z = (Z_1, \ldots, Z_K) \in \mathbb{R}^K$ the quantity $\Phi_\eta(Z) = \log\left(\sum_{k=1}^K e^{\eta Z_k}\right)/\eta$. A direct consequence of Taylor's theorem is that

$$\Phi_\eta(G_{n+1}) = \Phi(G_n) + \langle \nabla\Phi_\eta(G_n), g_{n+1}\rangle + \frac{g_{n+1}^T \nabla^2 G(\xi_n) g_{n+1}}{2},$$

where $g_{n+1} = \left(X_{n+1}^{(k)} - p_{m+1}X_{n+1}\right)_{k\in\mathcal{K}} \in \mathbb{R}^K$, $\nabla^2 G$ is the Hessian of G and ξ_n belongs to the line $[G_n, G_{n+1}]$. Simple computations and the fact that $p_{n+1} = \nabla\Phi_\eta(G_n)$ show that

$$\langle \nabla\Phi_\eta(G_n), g_{n+1}\rangle = \langle \nabla\Phi_\eta(G_n), X_{n+1}\rangle - p_{n+1}X_{n+1} = 0 \quad \text{and} \quad g_{n+1}^T \nabla^2 G(\xi_n) g_{n+1} \leq \eta.$$

We finally get, using a simple induction and the fact that $R'_n \leq \Phi_\eta(G_n)$:

$$R'_n \leq \Phi_\eta(G_n) \leq \Phi_\eta(0) + \frac{n\eta}{2} = \frac{\log(K)}{\eta} + \frac{n\eta}{2} \leq \sqrt{2n\log(K)},$$

with the choice of $\eta = \sqrt{2\log(K)/n}$, see Cesa-Bianchi and Lugosi [2006] for more details. As mentioned before, the constant can be improved, but at the cost of simplicity.

- The second way to interpret the exponential weight algorithm is in term of *smooth (or quantal) best-responses*. At first sight, one might be tempted to play at stage n the action with the highest cumulative rewards $\sum_{m=1}^n X_m^{(k)}$, as in the stochastic framework. Unfortunately, this would lead to a linear regret, mostly because of the irregularity of the *best response mapping* $Z = (Z_1, \ldots, Z_K) \mapsto \arg\max_k Z_k$. Therefore, regularizing or smoothing this mapping is necessarily and can be done as follows. Given $Z \in \mathbb{R}^K$, define the ε-smooth best response to Z (with respect to the entropy $h : \Delta(\mathcal{K}) \to \mathbb{R}$) as

$$p^*_\varepsilon(Z) := \arg\max_{p\in\Delta(\mathcal{K})} pZ - \varepsilon h(p).$$

  Typically, the mapping h has to be convex. For the specific choice of $h(p) = \sum_{k=1}^K p_k \log(p_k)$, i.e., when h is the Shannon entropy, then $p^\star_\varepsilon(Z) = \nabla\Phi_{1/\varepsilon}(Z)$. The associated algorithm to a specific choice of entropy function is usually called *follow the regularized leader* or *smooth fictitious play*, Fudenberg and Levine [1995], Hazan [2012].
- A third way to interpret this algorithm, and this one can be used to efficiently implement it, is the following. At each stage $n \in \mathbb{N}$ and for each action $k \in \mathcal{K}$, let $\zeta_n^{(k)}$ be a random perturbation of the reward, and consider the algorithm choosing the action with the highest perturbed reward, i.e.,

$$k_{n+1} = \arg\max_k \sum_{m=1}^n X_m^{(k)} + \zeta_n^{(k)}.$$

  Then if $\zeta_n^{(k)}$ are i.i.d., drawn accordingly to some Gumble distribution then it is easy to show that the probability that $k_{n+1} = k$ is proportional to $e^{c\sum_{m=1}^n X_m^{(k)}}$, where the value of c depends on the parameters of the Gumble distribution. The exponential weight algorithm can therefore be viewed as a *perturbation* of the best-response mapping. The associated algorithm is referred to as *follow the perturbed leader* or *stochastic fictitious play*, Fudenberg and Kreps [1993].

Each of these interpretations can be generalized into a different class of algorithms, based on the minimization of some potential function or any regularization/perturbation of the best response.

In case of bandit monitoring, the main idea is to apply the exponential weight algorithm with respect to the sequences $\widehat{X}_n^{(k)}$ of unbiased estimators of $X_n^{(k)}$ defined by

$$\widehat{X}_n^{(k)} = X_n^{(k)} \frac{\mathbb{1}\{k_n = k\}}{p_n^{(k)}}, \text{ since } \mathbb{E}[\widehat{X}_n^{(k)}] = \frac{X_n^{(k)}}{p_n^{(k)}}\mathbb{E}[\mathbb{1}\{k_n = k\}] = X_n^{(k)}.$$

2.2. Stochastic Approximations: a Useful Tool for a Large Class of Algorithms

We now present an effective technique to prove that a given random sequence $(p_n)_n$ generated by the decision maker admits no regret *in average* (or, stated otherwise, that regret increases sub-linearly), as in the stochastic or smooth fictitious play introduced above. As mentioned before, we focus on the full monitoring case, where p_{n+1} can depend on the whole past history as well as the stage number n (in which case his decision rule is called *time inhomogeneous*). However we restrict our analysis to the particular case where it only depends on the average realizations observed so far, i.e., $p_{n+1} = \mu_n \left(\frac{1}{n}\sum_{m=1}^{n} X_i\right)$ where μ_n is some function to be chosen. Note that the exponential weight algorithm with parameter η_n (where η_n can be either constant or time-dependent) corresponds to the map μ_n defined by

$$\mu_n(x^{(1)},...,x^{(K)}) = \left(\frac{e^{n\eta_n x^{(k)}}}{\sum_{\ell=1}^{K} e^{n\eta_n x^{(\ell)}}}\right)_{k\in\mathcal{K}}$$

In this section, we are interested into the *average regret* up to time n, i.e.,

$$\overline{R}'_n := \frac{1}{n}R'_n = \max_{k\in\mathcal{K}} \overline{X}_n^{(k)} - \overline{\rho}_n, \quad \text{where} \;\; \overline{\rho}_n := \frac{1}{n}\sum_{m=1}^{n} p_m X_m$$

In this setting, a strategy of the decision maker has average regret bounded by ε if, with probability one, the random sequence $(\overline{X}_n, \overline{\rho}_n)$ converges to the set

$$C_\varepsilon := \left\{(\overline{X},\overline{\rho}) \in \mathbb{R}^K \times \mathbb{R} : \;\max_{k\in\mathcal{K}} \overline{X}^{(k)} - \overline{\rho} \le \varepsilon\right\},$$

in the sense that the limit set of $(\overline{X}_n, \overline{\rho}_n)$ is contained in C_ε. When $\varepsilon = 0$, we will say that there is asymptotically no (average) regret.

Let $\mathcal{F}_n$ be the history up to stage n. The evolution of the sequences $(\overline{X}_n)_n$ and $(\overline{\rho}_n)_n$ can be written as

$$\overline{X}_{n+1}^{(k)} - \overline{X}_n^{(k)} - \frac{1}{n+1}\left(X_{n+1}^{(k)} - \mathbb{E}\left(X_{n+1}^{(k)} \mid \mathcal{F}_n\right)\right) = \frac{1}{n+1}\left(-\overline{X}_n^{(k)} + \mathbb{E}\left(X_{n+1}^{(k)} \mid \mathcal{F}_n\right)\right), \;\; k \in \mathcal{K},$$

$$\overline{\rho}_{n+1} - \overline{\rho}_n - \frac{1}{n+1}\left(p_{n+1}X_{n+1} - \mathbb{E}\left(p_{n+1}X_{n+1} \mid \mathcal{F}_n\right)\right) = \frac{1}{n+1}\left(-\overline{\rho}_n + \mathbb{E}\left(p_{n+1}X_{n+1} \mid \mathcal{F}_n\right)\right)$$

Hence, denoting ξ_{n+1} the martingale differences on the left-hand side and $Z_n := \mathbb{E}(X_{n+1} \mid \mathcal{F}_n)$, we obtain the stochastic difference equation

$$\left(\overline{X}_{n+1}, \overline{\rho}_{n+1}\right) - \left(\overline{X}_n, \overline{\rho}_n\right) - \frac{1}{n+1}\xi_{n+1} = \frac{1}{n+1}\left(-\overline{X}_n + Z_n, -\overline{\rho}_n + \left\langle \mu_n(\overline{X}_n), Z_n\right\rangle\right)$$

Intuitively the martingale difference $(\xi_n)_n$ should not affect too drastically the behavior of the random sequence $(\overline{X}_n, \overline{\rho}_n)$ in the long run. Thus using an appropriate time-rescaling through the map $m(t) := \min\{n \in \mathbb{N} : \;\sum_{m=1}^{n} 1/i \ge t\}$, the asymptotic behavior of the random sequence $(\overline{X}_n, \overline{\rho}_n)$ should be closely related to the solutions of the non-autonomous differential inclusion

$$\frac{d}{dt}(\overline{X}(t), \overline{\rho}(t)) \in F\left(t, (\overline{X}(t), \overline{\rho}(t))\right), \;\text{where}\;\; F(t, \overline{X}, \overline{\rho}) = \left\{(-\overline{X} + Z, -\overline{\rho} + \left\langle \mu_{m(t)}(\overline{X}), Z\right\rangle), \;\; Z \in \mathbb{R}^K\right\}. \qquad (4)$$

Taking into account the fact that Z_n is completely unknown to the decision maker and therefore could be anything, the image sets of F can be arbitrarily large. It is also worthwhile pointing out that the differential inclusion (4) is non-autonomous, which makes the qualitative analysis of the solutions tricky. For instance it is not clear how the concepts of *attractor* or *Lyapunov function* should be defined in these settings.

However, in the particular case where the decision rule is autonomous, i.e; $\mu_n(x) = \mu(x)$, so is the differential inclusion and these notions can be generalized very naturally (see Benaïm et al. [2005]). Also and more importantly, if one can exhibit a Lyapunov function ϕ for the differential inclusion (4) with respect to the set C_ε then it can be proved (under the right conditions) that the limit set of $(\overline{X}_n, \overline{\rho}_n)$ is almost surely contained in C_ε. This in turn proves that the average regret is asymptotically bounded by ε. This technique was introduced in Benaïm et al. [2006] in a game theoretical framework to prove that *smooth fictitious play strategies* are ε-consistent for small enough smoothing parameters. In our settings this result can be reformulated as follows: for any $\varepsilon > 0$, if the decision maker uses an exponential weight algorithm with parameter $\eta_n = \frac{\eta}{n}$ then the average regret is smaller than ε, provided η is large enough. In that case, notice that the decision rule of the decision maker is $\mu_n = \mu$, where

$$\mu(x^{(1)}, ..., x^{(K)}) = \left(\frac{e^{\eta x^{(k)}}}{\sum_{\ell=1}^{K} e^{\eta x^{(\ell)}}} \right)_{k \in \mathcal{K}}$$

and it can be proved that the map ϕ defined by

$$\phi(\overline{X}, \overline{\rho}) = \left(\max_{p \in \Delta(\mathcal{K})} \left\{ \langle p, \overline{X} \rangle - \frac{1}{\eta} \sum_{k \in \mathcal{K}} p^{(k)} \log p^{(k)} \right\} - \overline{\rho} \right)^{+}$$

verifies $\frac{d}{dt}\phi(\overline{X}(t), \overline{\rho}(t)) \leq \frac{1}{\eta} + Ke^{-t}$ and is therefore a Lyapunov function for C_ε, provided η is large enough.

In Benaïm and Faure [2013], this technique is developed to analyze more general decision rules of the form

$$\mu_n(x) = Q_h(n\eta_n x), \quad \text{where} \quad Q_h(y) := \arg \max_{p \in \Delta(\mathcal{K})} \{ < y, p > - h(p) \},$$

that is Q_h is the convex conjugate of some strongly convex function h. In particular, when h is the Shannon entropy: $h(p) = \sum_{k \in \mathcal{K}} p^{(k)} \log p^{(k)}$, we recover the exponential weight algorithm (see above and previous section).

A natural question is then "how should the parameter sequence $(\eta_n)_n$ be chosen so that the average regret asymptotically vanishes?" Intuitively, a necessary condition is to have $n\eta_n \to +\infty$ (if this sequence goes to zero for instance then the decision maker tends to play uniformly). However if the sequence $n\eta_n$ goes to infinity too fast then the algorithm behaves like a best response algorithm and, as it was already mentioned in previous section, the average regret does not vanish. The main result in Benaïm and Faure [2013] is that if $n\eta_n$ goes to infinity and $\eta_n = \mathcal{O}(n^{-\alpha})$ for some positive α then there is no average regret.

However, according to their qualitative nature, the stochastic approximations techniques used to obtain this result do not provide explicit upper bound for regret. In a recent paper, Kwon and Mertikopoulos [2014], a continuous-time approach is used to obtain explicit upper bound for regret this class of algorithm. Unsurprisingly, if $\eta_n = \mathcal{O}(n^{-\alpha})$ then the optimal rate is obtained for $\alpha = 1/2$ and the upper bound of the average regret is then of the order $n^{-1/2}$. The techniques employed to obtain this result are based on an idea introduced in Sorin [2009] to study the exponential weight algorithm in continuous time.

2.3. **Improved Rates of Convergence**

As we mentioned in Section 2.1, bounding the regret R'_n by $\mathcal{O}\big(n \log(K)\big)$ is optimal in the worst case. But, a natural question is what happens in better situations. It is possible to get simultaneously worst case guarantees and faster rate if the sequences of rewards are more favorable. For example, if the best decision receives large cumulative reward $X_n^\star = \max_{k \in \mathcal{K}} \sum_{m=1}^{n} X_m^{(k)}$ then the agent can achieve $\mathcal{O}\big(\log(K)\sqrt{n - X_n^\star}\big)$. Cesa-Bianchi et al. [2007] raised the question whether it was possible to improve even further by proving second-order (variance like) bounds on the regret. They proved a bound in $\mathcal{O}\Big(\log(K)\big(\sum_{m=1}^{N} v_m\big)^{1/2}\Big)$, where $v_m = \mathrm{Var}_{p_m}\big[X_m^{(k_m)}\big]$ is the variance of the agent's forecast at stage m. This bounds has its advantages, however its uniformity over all decisions does not reflect that it may be easier to compete with some stable decision than with others. Besides, one may find unpleasant that the bound depends on the forecasts of the agent itself.

To understand the intrinsic hardness of the problem and retrieve the classical biais-variance trade-off, Cesa-Bianchi et al. [2007] also suggested the following desirable bound on the cumulative reward

$$\sum_{m=1}^{n} p_m X_m \geq \max_{k \in \mathcal{K}} \left\{ \sum_{m=1}^{n} X_m^{(k)} \; + \; \square \sqrt{\log(K) \sum_{m=1}^{n} \left(X_m^{(k)} - \overline{X}_n^{(k)} \right)^2} \right\} , \tag{5}$$

where $\square$ denotes some universal constant. Achieving this bound remains an open problem. Various methods come close to accomplishing (5) but in each of these works the regret bound for agent k (the right term in (5)) also depends on rewards of other agents (see Chiang et al. [2012], Gaillard et al. [2014], Hazan and Kale [2010]).

However, such improved bounds still deserve consideration. We display below the one of Gaillard et al. [2014], which expresses the regret in terms of excess gains,

$$\sum_{m=1}^{n} p_m X_m \geq \max_{k \in \mathcal{K}} \left\{ \sum_{m=1}^{n} X_m^{(k)} \; + \; \square \sqrt{\log(K) \sum_{m=1}^{n} \left(X_m^{(k)} - p_m X_m \right)^2} \right\} . \tag{6}$$

This is achieved by a variant of the prod forecaster of Cesa-Bianchi et al. [2007]. The main idea is to consider a different learning rate for each decision. It forms at stage m the weight vector, defined component-wise by

$$p_m^{(k)} = \frac{\eta_k \prod_{t=1}^{m-1} \left(1 + \eta_k \big(X_t^{(k)} - p_t X_t\big)\right)}{\sum_{l \in \mathcal{K}} \eta_l \prod_{t=1}^{m-1} \left(1 + \eta_l \big(X_t^{(l)} - p_t X_t\big)\right)} , \tag{7}$$

where $(\eta_k)_{k \in \mathcal{K}}$ are learning parameters that can be calibrated online (to a small multiplicative cost of $\log \log n$).

Regret bound of form (6) present several nice features. For instance, assume that the sequences of rewards $\big(X_m^{(k)}\big)$ are no longer adversarial, but are i.i.d. Assume also that one decision is significantly better than others, then any algorithm that achieves Equation (6) only suffers constant regret with high probability, as in the benchmark framework. Wintenberger [2014] also proves Equation (6) for a variant of the exponentially weighted average forecaster and derives nice consequence of (6) in a generic stochastic setting. More precisely, he obtains optimal bounds on the predictive risk. Another nice application of Equation (6) also occurs when a decision receives large rewards, and another one in the context of experts reporting there confidence introduced by Blum and Mansour [2007].

2.4. **Generalized Regret and Generalized Exponential Weight Algorithm**

In the adversarial framework, the notion of regret can be refined into more precise concepts. For instance, the *internal regret*, see Blum and Mansour [2007], Foster and Vohra [1999], is defined as the usual regret on the set of stages where a specific action was chosen. Mathematically, it reads as follows

$$R_n^i := \max_{\ell \in \mathcal{K}} \max_{k \in \mathcal{K}} \left\{ \sum_{m : k_m = \ell} X_m^{(k)} - \sum_{m : k_m = \ell} X_m^{(k_m)} \right\} = \max_{\ell \in \mathcal{K}} \max_{k \in \mathcal{K}} \left\{ \sum_{m : k_m = \ell} X_m^{(k)} - \sum_{m : k_m = \ell} X_m^{(\ell)} \right\} .$$

Informally, having no internal regret means that, on the set of stages where the action $\ell \in \mathcal{K}$ was played, it was the best constant action to play. The concept of internal regret can even be more refined in *swap regret*. The main idea is that the strategy of the decision maker must be (asymptotically and in average) at least as good as the strategy that plays action $\phi(k)$ when the decision maker chooses k; such a mapping $\phi(\cdot)$ is called a *swap-mapping* (the usual regret correspond to the case where the swap mappings considered are all constant). The swap regret is therefore defined as

$$R_n^s := \max_{\phi:\mathcal{K}\to\mathcal{K}} \left\{ \sum_{m=1}^n X_m^{(\phi(k_m))} - \sum_{m=1}^n X_m^{(k_m)} \right\}.$$

Similarly to usual regret, swap regret can equivalently be written in expectation as

$$R_n^{s'} = \max_{\phi:\mathcal{K}\to\mathcal{K}} \left\{ \sum_{m=1}^n \sum_{k=1}^K p_m^{(k)} X_m^{(\phi(k))} - \sum_{m=1}^n p_m X_m \right\} = \max_{\phi:\mathcal{K}\to\mathcal{K}} \left\{ \sum_{m=1}^n p_m[\phi] X_m - \sum_{m=1}^n p_m X_m \right\},$$

where $p_m[\phi]^{(k)} = \sum_{\ell:\phi(\ell)=k} p_m^{(\ell)}$.

Even more generally, we can define a general adaptive swap mapping as any function ξ that maps to any finite history a specific action, i.e.,

$$\xi : \bigcup_{n\in\mathbb{N}} \left\{ p_m, X_m^{(k)}; m \le n, k \in \mathcal{K} \right\} \to \mathcal{K}.$$

Given a finite set Ξ of such adaptive mappings, we define the associated Ξ-regret as

$$R_N^{\Xi'} = \max_{\xi\in\Xi} \left\{ \sum_{m=1}^n \sum_{k=1}^n p_m^{(k)} X_m^{(\xi(h_m^{(k)}))} - \sum_{m=1}^n p_m X_m \right\}, \quad \text{where} \quad h_m^k = \left\{ p_s, k, X_s^{(\ell)}; s \le m-1, \ell \in \mathcal{K} \right\}.$$

A key result is that, as soon as Ξ is finite, regret can be sub-linear, Perchet [2014]. Indeed, introduce the potential Φ_η^Ξ from $\mathbb{R}^\Xi$ into $\mathbb{R}$ defined, as before, by $\Phi_\eta^\Xi(Z) = \log(\sum_{\xi\in\Xi} e^{\eta Z_\xi})/\eta$. Taylor's theorem yields again

$$\Phi_\eta^\Xi(G_{n+1}^\Xi) = \Phi_\eta^\Xi(G_n) + \langle \nabla\Phi_\eta^\Xi(G_n), g_{n+1}^\Xi \rangle + \frac{(g_{n+1}^\Xi)^T \nabla^2 \Phi_\eta^\Xi(\zeta_n) g_{n+1}^\Xi}{2},$$

where $G_n^\Xi = \sum_{m=1}^n \sum_{k=1}^n p_m^{(k)} X_m^{(\xi(h_m^{(k)}))} - \sum_{m=1}^n p_m X_m$ and $g_{n+1}^\Xi = G_{n+1}^\Xi - G_n^\Xi$ and $\zeta_n \in [G_n^\Xi, G_{n+1}^\Xi]$. To apply the same argument as before, it only remains to prove that there always exists p_{n+1} such that the inner product is non-positive, no matter the choice of $X_{n+1}^{(k)}$. This is actually a consequence of von Neumann minmax theorem. Indeed, $\langle \nabla\Phi_\eta^\Xi(G_n), g_{n+1}^\Xi \rangle$ is linear both in p_{n+1} and in $\{X_n^{(k)}\}_k$, as a consequence

$$\min_{p_{n+1}\in\Delta(\mathcal{K})} \max_{X_{n+1}^{(k)}\in[0,1]^K} \langle \nabla\Phi_\eta^\Xi(G_n), g_{n+1}^\Xi \rangle = \max_{X_{n+1}^{(k)}\in[0,1]^K} \min_{p_{n+1}\in\Delta(\mathcal{K})} \langle \nabla\Phi_\eta^\Xi(G_n), g_{n+1}^\Xi \rangle. \tag{8}$$

As a consequence, one just has to check that for any $\{X_n^{(k)}\}_k \in [0,1]^K$ there exists p such that the inner product is non-positive. This is immediate by taking $p = e_{k^*}$ where $k^* \in \arg\max_k X_n^{(k)}$.

As a consequence, choosing p_{n+1} as a solution of the minmax problem (8) ensures that

$$R_n^{\Xi'} \le \sqrt{2\log(\Xi)n}.$$

Specific examples includes external, internal and swap regret, where the set of mappings is of size at most K, K^2 and K^K, therefore with respective upper bounds of $\sqrt{2\log(K)n}$, $2\sqrt{\log(K)n}$ and $\sqrt{2K\log(K)n}$.

2.5. **Application: Load Forecasting**

In this section, we are concerned with the application of short-term (one day ahead) forecasting of the electricity consumption. For more details, the reader is referred to the empirical study of Gaillard and Goude [2014]. Load forecasting is vitally important for electric providers like Electricité De France (EDF). It helps to make decisions on purchasing and generating electric power so as to maintain the equilibrium between production and demand. A variety of methods and ideas have been explored in the past decades. This makes it especially interesting for our setting of prediction with expert advice.

We consider a load data set which includes half-hourly observations of the total electricity consumption of the EDF market in France from January 1, 2008 to June 15, 2012, together with several covariates, including temperature, cloud cover, wind, etc. We aim at forecasting the consumption every day at 12:00 for the next 24 hours; that is, for the next 48 time instances. This data set was analyzed in Devaine et al. [2013], Gaillard and Goude [2014]. The data is partitioned into two pieces: a training set from January 1, 2008 to August 31, 2011 to fit the forecasting methods; a testing set from September 1, 2011 to June 15, 2012 to evaluate their performance and to perform the sequential aggregation methods. Operational forecasting purposes require the predictions to be made simultaneously at 12:00 for the next 24 hours. Aggregation rules can be adapted to this constraint via a generic extension detailed in Devaine et al. [Devaine et al., 2013, Section 5.3].

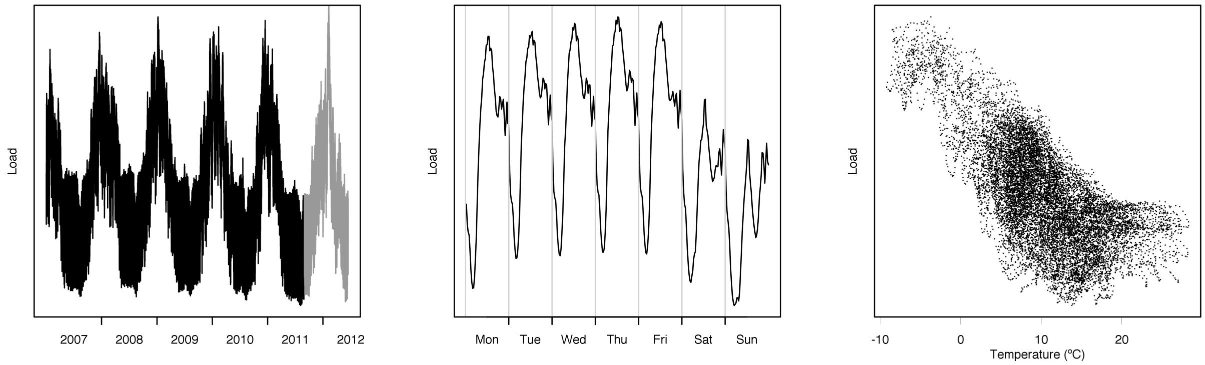

FIGURE 1. [left] The observed half-hourly electricity consumptions between January 1, 2008 to June 15, 2012. Approximately the last year (in gray) is used to test the methods. [middle] The observed half-hourly electricity consumptions during a typical week. A weekly pattern can be observed with a reduction of consumption during the week-end. [right] The impact of temperature on electric load.

Our choice of individual models includes a set of three forecasting methods: the GAM forecaster captures non-linear relationships between electric load and different covariates (temperature, nebulosity,...) by performing a regularized linear regression on a spline basis transfer functions (see Pierrot and Goude, 2011, Pierrot et al., 2009); the CLR forecaster performs a data driven dimension reduction together with a data transformation so as to reduce the problem to a simple linear regression (see Cho et al., 2013, 2014); the KWF is a nearest neighbors approach on a wavelet basis (see Antoniadis et al., 2006, 2010, 2012, 2013). These individual forecasters were carefully chosen because they exhibit good performance and various behaviors.

We consider three combining algorithms: the exponentially weighted average forecaster (EWA) introduced in learning theory by Littlestone and Warmuth [1994], Vovk [1990] and presented in Section 2.1; the ridge regression forecaster (Ridge) introduced in the context of prediction with expert advice by Azoury and Warmuth [2001]; and the prod forecaster with multiple learning rates (ML-Prod) introduced in Gaillard et al. [2014] and recalled in Equation (7). The learning parameters of EWA and Ridge were optimized online on adaptive finite grids as suggested in [Devaine et al., 2013, Section 2.4]. The learning rates of ML-Prod were theoretically calibrated according to Gaillard et al. [2014]. Ridge aims at competing with the best linear combination oracle, while EWA and ML-Prod compete with the best convex combination (non-negative weights that sum to 1) by resorting to the gradient trick (see [Cesa-Bianchi and Lugosi, 2006, Section 2.5]).

A sequence of prediction $\hat{y}_1^n = (\widehat{y}_1, \dots, \widehat{y}_n)$, formed over the testing set by an individual forecaster or by a combining algorithm, is evaluated by its root mean square error (RMSE) and by its absolute percentage of error (MAPE), respectively defined by:

$$\textsc{rmse}(\hat{y}_1^n) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \widehat{y}_t)^2} \qquad \text{and} \qquad \textsc{mape}(\hat{y}_1^n) = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{y_t}\,,$$

where $y_1^n = (y_1, \dots, y_n)$ denotes the half-hourly sequence of observations of the load along the testing set ($n = 48 \times 244$ days $= 11\,712$ time steps).

Methods	RMSE (MW)	MAPE (%)
Best expert	744	1.29
Best convex combination	629	1.06
Best linear combination	629	1.06
EWA	624	1.07
ML-Prod	625	1.04
Ridge	638	1.06

Figure 2. Performance of oracles and combining algorithms.

Table 2 reports the errors obtained by the combining algorithms and by three oracles (the best individual forecaster, the best convex combination of individual forecasters, and the best linear combination). The MAPEs are around 1%. We observe similar results for the best convex combination and for the best linear combination with RMSEs of 629 MW. This explains why Ridge does not perform better than other combining algorithms: there is no visible gain in competing against the best linear combination, but the estimation cost is larger. The best individual forecaster gets 744 MW. This motivates the necessity of combining these models which bring different information. Remark that the effective regrets of EWA and ML-Prod are actually negative. This shows how pessimistic the worst case regret bound $O\big(n\log(K)\big)$ is in real world. They are indeed stated in the worst case scenario, which is unlikely to occur in reality.

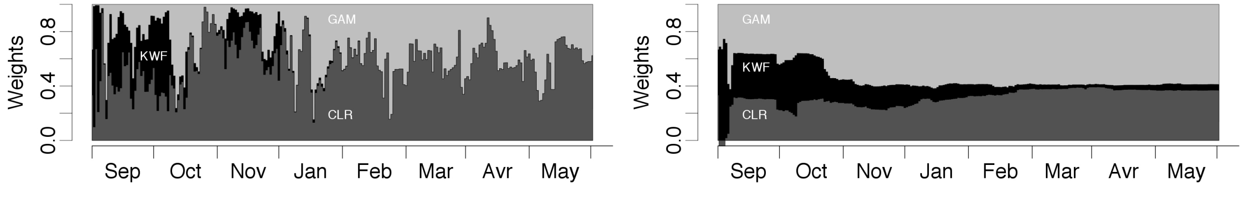

Figure 3. Time evolution of the weights assigned to the three individual forecaster by ML-Prod [left] and by Ridge [right].

Figure 3 displays the time evolution of the weights assigned by ML-Prod and by Ridge during the testing period. Ridge forms weights that are much are much more stable. Stability may be desirable if one ask for robust methods that can predict at larger horizons of time as well. On the other hand, it is less adaptive to a changing environment.

3. Learning in Games

From now on, we shall assume that the sequence of rewards $X_n^{(k)}$ are no longer chosen adversarially by Nature that has no objectives, reason or motivations but rather that there are a finite number of players whose repeated choices of actions induce sequence of rewards to them.

3.1. **A Reminder of Static/Dynamic Game Theory**

First of all, we recall some basic notions of game theory. Assume that there is a finite number of players $i = 1, \ldots, I$ and that player i chooses action k_i in a finite set $\mathcal{K}_i$. The choices of actions $k_1, \ldots, k_I$ generate a reward to player i through the payoff mapping $u^{(i)}(k_1, \ldots, k_I)$. As mentioned above, players might want to choose actions at random (or *play a mixed action*), accordingly to $p_i \in \Delta(\mathcal{K}_i)$; in that case, the associated expected payoff for player i is

$$u^{(i)}(p_1, \ldots, p_I) := \sum_{(k_1,\ldots,k_I)\in\prod\mathcal{K}_i} \prod_j p_j^{(k_j)} u^{(i)}(k_1, \ldots, k_I) = \mathbb{E}_{p_1\otimes\ldots\otimes p_I}[u^{(i)}].$$

The following notation is widely used and quite useful

$$u^{(i)}(p_{-i}, q_i) = (p_1, \ldots, p_{i-1}, q_i, p_{i-1}, \ldots, p_I)$$

Among all the solution concepts of game theory, Nash equilibrium is probably the most important. The hindsight is that a point $(p_1^\star, \ldots, p_I^\star)$ is a Nash equilibrium if all player i, assuming the others are going to play accordingly to $p_1^\star$, $p_2^\star$, etc., has no strict interest to play something else than $p_i^\star$. Stated otherwise, if all other players play $p_j^\star$, then player i has no interest in *deviating* from playing $p_i^\star$. Formally, $(p_1^\star, \ldots, p_I^\star)$ is a Nash equilibrium if

$$u^{(i)}(p_{-i}^\star, p_i^\star) = \max_{p\in\Delta(\mathcal{K}_i)} u^{(i)}(p_{-i}^\star, p), \quad \forall i \in \{1, \ldots, I\}.$$

An interesting feature of Nash equilibria is that they always exist (at least in mixed actions) when all action sets are finite. Maybe more surprisingly, there are generically an odd number of them, and it is complex to compute them (see, e.g., von Stengel [2002]).

This is one of the reasons why correlated equilibria have been introduced. Instead of playing independently (as in Nash equilibria), we can assume that players can correlate their choices of actions using some external device that picks $(k_1, \ldots, k_I) \in \prod \mathcal{K}_i$, accordingly to some known product distribution $q^* \in \Delta(\prod \mathcal{K}_i)$, and such that player i only observes k_i (and not the other coordinates k_{-i}). The product distribution $q^* \in \Delta(\prod \mathcal{K}_i)$ is a correlated equilibrium if for all player $i \in \{1, \ldots, I\}$ and all action $k \in \mathcal{K}_i$ having a positive probability of being played, the player i has no incentive to deviate by playing $\ell \in \mathcal{K}_i$ instead, i.e., if

$$u^{(i)}(q_{-i}^\star[k], k) = \max_{\ell\in\mathcal{K}_i} u^{(i)}(q_{-i}^\star[k], \ell),$$

where $q_{-i}^\star[k]$ is the marginal of q conditional to k. Note that this immediately reduces (up to a multiplication of both sides by 0) to

$$\sum_{k_{-i}} q^\star(k_{-i}, k) u^{(i)}(k_{-i}, k) \leq \sum_{k_{-i}} q^\star(k_{-i}, k) u^{(i)}(k_{-i}, \ell), \quad \forall i \in \{1, \ldots, I\}, \forall k, \ell \in \mathcal{K}_i\,.$$

So correlated equilibria are defined by a finite number of linear inequalities. The set of all correlated equilibria is therefore a convex compact polytope and is simpler to compute than the set of Nash equilibria (which is by definition a subset of the set of correlated equilibria).

There are special classes of games of specific interest:

Zero-Sum Games: These games are two player games such that $u^{(1)} = -u^{(2)}$, i.e., the gain of a player is the loss of the other. A key property of these games is that players have *optimal strategies* that guarantee them to obtain at least some reward, called *the value of the game*, no matter what their opponent is playing. In comparison, in non-zero sum games, the mixed action to be played by a player at a Nash equilibrium depends on the actions of his opponents.

These statement are summarized in the minmax theorem of von Neumann:

$$\max_{p_1^\star \in \Delta(\mathcal{K}_1)} \min_{p_2 \in \Delta(\mathcal{K}_2)} u^{(1)}(p_1^*, p_2) = \min_{p_2^* \in \Delta(\mathcal{K}_2)} \max_{p_1 \in \Delta(\mathcal{K}_1)} u^{(1)}(p_1, p_2^\star)$$

and the value of the game is precisely the common value of these problems. Optimal actions for player 1 are any point $p_1^\star$ that maximizes $\min_{p_2 \in \Delta(\mathcal{K}_2)} u^{(1)}(\cdot, p_2)$ and similarly for player 2.

In zero-sum games, players should always play an optimal action, since this ensures them at least the value and any other choice can give them a strictly smaller reward. So given a zero-sum game, players have some well defined "optimal" actions. This is not the case in non-zero sum games, even if the set of Nash equilibria is given. Indeed, they are typically non-unique, with a finite number of them. So knowing the set of equilibria is not sufficient to know the action a player should play, since the prescribed action at a given equilibria might be suboptimal for another one. Player must *learn* somehow which equilibria is going to be played before playing it.

Potential Games: Those games, introduced by Rosenthal [1973], have an additional assumption on the reward mappings. We assume that there exists a *potential function* $\phi : \prod \mathcal{K}_i \to \mathbb{R}$ such that

$$u^{(i)}(k_{-i}, k) - u^{(i)}(k_{-i}, \ell) = \phi(k_{-i}, k) - \phi(k_{-i}, \ell), \quad \forall i \in \{1, \ldots, I\}, k, \ell \in \mathcal{K}_i.$$

Here, the crucial assumption is that ϕ does not depend on i. There always exists a *pure* Nash equilibrium (i.e., an equilibrium where any player plays a pure action) in those games: any maximizer of the function ϕ. Computing them can therefore be tractable.

In the model we consider, we assume that each player i chooses sequentially actions $k_{i,n} \in \mathcal{K}_i$ and gets the sequence of payoff $u^{(i)}(k_{1,n}, \ldots, K_{I,n})$. Even if they know that there are several players choosing actions simultaneously, we assume that they do not known how many opponents they have, what are their actions sets and payoff mappings: the only observation available to player i at stage n is his current payoff and, possibly, the payoff he would have got had he played another action. In particular, we assume that players do not try to, or at least can not, infer the parameters of the games (the set $\mathcal{K}_j$ and/or the mappings $u^{(j)}$) but rather whether their individual and *uncoupled* behaviors (i.e., the decisions taken by a player do not depend on the payoff mappings of other players) converge, in some sense to be defined later, to the set of Nash equilibria or any other solution concepts.

We might consider three different possible definitions of convergence or *learning Nash equilibria*, from the strongest to the weakest.

i) The first one requires that the profile of mixed action $\vec{p}_n = (p_{1,n}, \ldots, p_{I,n}) \in \prod \Delta(\mathcal{K}_i)$, where $p_{i,n} \in \Delta(\mathcal{K}_i)$ is the mixed action used by player i at stage n, converges to the set of Nash equilibria. We emphasize here that we do not require convergence to one specific Nash equilibrium, but convergence to the whole set, in the sense that the distance from $\vec{p}_n$ to the set goes to 0.

ii) The second convergence is in average. Players learn Nash equilibria if $\vec{\overline{p}}_n = (\overline{p}_{1,n}, \ldots, \overline{p}_{I,n}) \in \prod \Delta(\mathcal{K}_i)$, where $\overline{p}_{i,n} = \sum_{m=1}^n p_{i,m}/n$ or more or less equivalently $\overline{p}_{i,n} = \frac{1}{n}\sum_{m=1}^n \delta_{k_{i,m}}$, converges to the set of Nash equilibria.

iii) The last type of convergence happens if the joint empirical distribution $\frac{1}{n}\sum_{m=1}^n \vec{p}_m \in \Delta(\prod \mathcal{K}_i)$ converges to the set of Nash equilibria; a more or less equivalent requirement is that $\frac{1}{n}\sum_{m=1}^n \delta_{k_{1,m},\ldots,k_{I,m}}$ converges to the set of Nash equilibria.

Learning a different class of equilibria (say, correlated equilibria) is defined in a similar way.

3.2. **Classes of Learnable Equilibria**

Unfortunately, Nash equilibria are not *learnable* by any uncoupled strategies. No matter the profile of strategies, there always exists at least one game in which none of three aforementioned convergence occurs, see Hart and Mas-Colell [2003, 2006].

For instance, if all players minimize their regret then $\sum_{m=1}^n \vec{p}_m/n \in \Delta(\prod \mathcal{K}_i)$ converges to the Hannan set that contains all probability distribution $q \in \Delta(\prod \mathcal{K}_i)$ such that

$$\mathbb{E}_q u^{(i)}(k_1, \ldots, k_I) \geq \max_{k_i^* \in \mathcal{K}_i} u^{(i)}(k_i^*, q_{-i}),$$

where q_{-i} is the marginal of q on the coordinates different from i. This statement does not need any proof as it is a direct consequence of the definition of regret minimization. The Hannan set can actually be defined as the limit behavior of regret minimization strategies.

In special subclasses of games though, convergence to Nash equilibria might occur. One of the main striking example are zero-sum games. Indeed, $q \in \Delta(\mathcal{K}_1 \times \mathcal{K}_2)$ belongs to the Hannan set if and only if

$$\min_{p_2 \in \Delta(\mathcal{K}_2)} \max_{k_1 \in \mathcal{K}_1} u^{(1)}(k_1, p_2) \leq \max_{k_1 \in \mathcal{K}_1} u^{(1)}(k_1, q_{-1}) \leq \mathbb{E}_q[u^{(i)}(k_1, k_2)]$$

and similarly, since $u^{(2)} = -u^{(1)}$,

$$\mathbb{E}_q[u^{(i)}(k_1, k_2)] \leq \min_{k_2 \in \mathcal{K}_2} u^{(1)}(q_{-2}, k_2) \leq \max_{p_1 \in \Delta(\mathcal{K}_1)} \min_{k_2 \in \mathcal{K}_2} u^{(1)}(p_1, k_2).$$

As a consequence $(\overline{p}_{1,n}, \overline{p}_{2,n})$ converges, in the second sense, to the set of optimal strategies and, similarly, the average payoff converges to the value of the game. The existence of regret minimization strategies actually proves von Neumann's minmax theorem, see, e.g., Cesa-Bianchi and Lugosi [2006]. Indeed, given any strategies of both players, denote by $R_n^{(1)}$ and $R_n^{(2)}$ the respective regret of player 1 and 2 at stage n. Then it holds that

$$\min_{p_2 \in \Delta(\mathcal{K}_2)} \max_{k_1 \in \mathcal{K}_1} u^{(1)}(k_1, p_2) \leq \frac{1}{n} R_n^{(1)} + \frac{1}{n} \sum_{m=1}^n u^{(1)}(p_{1,m}, p_{2,m}) \leq \max_{p_1 \in \Delta(\mathcal{K}_1)} \min_{k_2 \in \Delta(\mathcal{K}_2)} u^{(1)}(p_1, k_2) + \frac{1}{n} R_n^{(1)} + \frac{1}{n} R_n^{(2)}.$$

As a consequence, the existence of strategies with sublinear linear regret ensures that

$$\min_{p_2 \in \Delta(\mathcal{K}_2)} \max_{p_1 \in \Delta(\mathcal{K}_1)} u^{(1)}(p_1, p_2) \leq \max_{p_1 \in \Delta(\mathcal{K}_1)} \min_{p_2 \in \Delta(\mathcal{K}_2)} u^{(1)}(p_1, p_2).$$

The other inequality always holds, this is therefore an equality which entails the minimax theorem.

We conclude this section by mentioning that although Nash equilibria are not learnable, correlated equilibria are. Indeed, assume that all players minimize their internal regret, without taking into account that they are facing other players. Then the empirical distribution of actions converges, in the third sense, to the set of correlated equilibrium.

Indeed, the internal regret $R_n^{(i)}$ of player i satisfies, for all $k^\star \in \mathcal{K}_i$

$$\max_{\ell \in \mathcal{K}_i} \sum_{m : k_m^{(i)} = k^\star} u^{(i)}(k_{m,-i}, \ell) - \sum_{m : k_m^{(i)} = k^\star} u^{(i)}(k_{m,-i}, k^\star) \leq R_n^{(i)}$$

where $k_{m,-i}$ is the profile of actions played by the other players at stage m. Defining $q_n = \frac{1}{n}\sum_{m=1}^{n} \delta_{(k_{1,m},\ldots,k_{I,m})}$, this inequality becomes

$$\max_{\ell \in \mathcal{K}_i} u^{(i)}(q_{n,-i}[k^\star], \ell) - u^{(i)}(q_{n,-i}[k^\star], k^\star) \leq \frac{R_n^{(i)}}{n} \frac{1}{q_{n,i}[k^\star]},$$

where $q_{n,i}[k^\star] = \frac{1}{n}\sharp\{m : k_m^{(i)} = k^\star\}$ and $q_{n,-i}[k^\star]$ is the empirical mixed action played by other players when player i had chosen action $k^\star$. Consider any "internal regret minimizing strategies", i.e., such that $R_n^{(i)}/n \to 0$, and $q^\star$ any accumulation point of $\{q_n\}$. Then if $q_i^\star[k] > 0$ we necessarily get by continuity

$$\max_{\ell \in \mathcal{K}_i} u^{(i)}(q_{-i}^\star[k], \ell) = u^{(i)}(q_{-i}^\star[k], k),$$

thus $q^\star$ is a correlated equilibrium.

3.3. Learning (Asynchronously) Nash Equilibria in Potential Games

Potential games have proven very useful, especially in the context of routing games, first mentioned in Beckman et al. [1956] and exhaustively studied ever since, in the transportation as well as computer science literature, see for example Gallager [1977], Orda et al. [1993], Wardrop [1954] and for distributed optimization (see for example Roughgarden [2005]).

As mentioned before, potential games always admit at least one pure Nash equilibrium, any maximizer of the potential function ϕ. This also provides a framework to design a learning algorithm.

We consider the *best response correspondence* $BR_i(k)$ as the set of all actions that maximize the payoff for player i under profile k:

$$BR_i(k) \equiv \left\{ \operatorname*{argmax}_{\alpha} u^i(\alpha; k_{-i}) \right\}. \tag{9}$$

A *Nash equilibrium* (NE) is a fixed point of the correspondence, i.e., a profile k^* such that $k_i^* \in BR_i(k^*)$ for every player i. This provides a distributed learning algorithm of Nash that consists in iterating the best response for all the players until convergence.

```
Best Response Algorithm (BRA):
        Repeat until no player changes its action
          1. Pick player $i$ at random
          2. Select new action $k_{i,t+1} := BR_i(u^{(i)}(t))$
```

Just checking that the potential increases whenever a player updates its action in Algorithm (BRA) yields that for any potential game, Algorithm (BRA) converges in finite time, almost surely, to a Nash equilibrium, see Monderer and Shapley [1996].

However, this learning algorithm suffers from several drawbacks.

1) First, using best response requires that each player has exact knowledge of the payoff of all its actions.
2) Letting players play simultaneously instead on one after the other destroys the convergence property. So that time coordination between players is needed unless the *revision sets* (sets of players that can play simultaneously) have a separation property (see Coucheney et al. [2014a]).
3) Finally, this algorithm is not robust to estimation errors on the payoffs.

A natural idea to fix these problems is to replace the greedy decisions of the players by a smoothed randomized best response map. Player i mixes its actions with distribution p_i, as it was done in Section 2.2:

$$p_i(u^{(i)}) = \arg\max_{p \in \Delta(\mathcal{K}_i)} \left\{ \sum_{\beta} p^{\beta} u^{(i)}(\beta) - h_i(p) \right\}, \tag{10}$$

where h_i is a smooth strongly convex function, steep on the boundary of the simplex, which acts as a *penalty* to the expected payoff $\sum_{\beta} p[i]^{\beta} u^{(i)}(\beta)$ of player i.

Convergence to a global maximizer of ϕ (hence, a Nash equilibria) of the modified algorithm can be proved using the theory of non-homogeneous perturbed Markov chains, see Coucheney et al. [2014a]. However, using a smoothed best response map does not help in solving the problems of the previous algorithm: the modified algorithm still requires the same information for each player (payoffs of all its actions) and the necessary and sufficient condition on the revision set for convergence are the same as for best response, see Coucheney et al. [2014a].

To solve these problems, one has to get away from the Markovian nature of the previous algorithms. Here are the main two ingredients of the upcoming algorithm:
First it keeps track of all past actions payoffs, through their *score* (discounted sum of all their past rewards), and second, it uses the previous smoothed best response, but on the current scores of the actions, and not on the payoffs directly.

One of the nice features of this approach actually lies in the proof of its convergence. The proof (sketched below) is based on the construction of a continuous dynamic for which we prove a form of the folk theorem of convergence to equilibria and stochastic approximation techniques to design a discrete learning algorithm.

In continuous time, the score of action α for player i at time t follows a classical discounting scheme :

$$y_{k,i}(t) = \int_0^t e^{-T(t-s)} u^{(i)}(k, p_{-i}(s))\, ds, \tag{11}$$

or, in differential form:

$$\dot{y}_{k,i} = u^{(i)}(k) - T y_{k,i}, \tag{12}$$

where T denotes the model's *discount rate* and $p_{-i}(s) \in \mathcal{X}$ is the other players' mixed strategy at time s.

As for players' choices, they use the same smoothed best response as before. To make equations more explicit, let us consider the case where h is the entropy: $h(x) = \sum_{\beta} p^{\beta} \log p^{\beta}$. This yields the so-called *logit map* for p:

$$p_{k,i}(y) = \frac{\exp(y_{k,i})}{\sum_{\ell} \exp(y_{\ell,i})}, \tag{13}$$

.

Combining the score and strategy dynamics gives a *penalty-regulated learning process* as an adjusted replicator equation, see Coucheney et al. [2014b]:

$$\dot{p}_{k,i} = p_{k,i} \left[u^{(i)}(k,p) - \sum_{\ell} p_{\ell,i} u^{(i)}(\ell,p) \right] - T\, p_{k,i} \left[\log p_{k,i} - \sum_{\ell} p_{\ell,i} \log p_{\ell,i} \right]. \tag{14}$$

As the name implies, when the discount rate T vanishes, (14) freezes to the ordinary (asymmetric) replicator dynamics of Taylor and Jonker [1978].

When the game has potential Φ, the dynamical system 14 admits a Lyapounov function

$$F(p) \equiv T \sum_i h_i(p_i) - \phi(p). \tag{15}$$

This readily characterizes the asymptotic behavior of (14) under the form of a folk theorem: For $T > 0$, if $q \in \mathcal{X}$ is Lyapunov stable then it is also a quantal response equilibria (QRE) of $\mathfrak{G}$ with rationality level $1/T$.

Here $q = (q_1, \dots, q_I)$ is a QRE if, for some $\rho \geq 0$ and for all $i \in \mathcal{N}$:

$$q_{k,i} \equiv \frac{\exp(\theta u^{(i)}(k, q_{-i}))}{\sum_{\ell} \exp(\theta u^{(i)}(\ell, q_{-i}))}.$$

The scale parameter $\theta \geq 0$ will be called the *rationality level* of the QRE in question. Obviously, when $\theta = 0$, QRE have no ties to the game's payoffs; on the other hand, when $\theta \to \infty$, quantal response functions approach best responses and QRE tend to become NE.

3.4. **Distributed Learning Algorithm**

Let us now examine how the dynamical system (14) can be used to design a new learning algorithm in finite games that are played repeatedly over time:

Penalized Learning Algorithm (PLA):

1. At stage $n+1$, each player selects an action $k_i(n+1)$ based on a mixed strategy $p_i(n) \in \Delta(\mathcal{K}_i)$.
2. Every player receives a bounded and unbiased estimate $\hat{u}_k^{(i)}(n+1)$ of his actions' payoffs, viz.
   - (a) $\mathbb{E}\left[\hat{u}_k^{(i)}(n+1) \,\middle|\, \mathcal{F}_n\right] = u_k^{(i)}(p_{-i}(n))$,
   - (b) $\left|\hat{u}_k^{(i)}(n+1)\right| \leq C$ (a.s.),

   where $\mathcal{F}_n$ denotes the history of the process up to stage n and $C > 0$ is a fixed constant.
3. Every player updates the score $Y_{k,i}(n+1)$ of each action according to a discretization of (14): $Y_{k,i} \leftarrow Y_{k,i} + \gamma_n(\hat{u}_k^{(i)}(n+1) - TY_{k,i})$;
4. Players choose a new mixed strategy $p_i(n+1) = logit(Y_i)$ and the process repeats ad infinitum.

The fact that the conditional expectations satisfy

$$\mathbb{E}\left[(Y_{k,i}(n+1) - Y_{k,i}(n))/\gamma_{n+1} \mid \mathcal{F}_n\right] = u^{(i)}(k, p(n)) - TY_{k,i}(n)$$

makes this algorithm a *stochastic approximation* of the penalty-regulated dynamics.

The theory of stochastic approximations, see Section 2.2 or Benaïm [1999], can be used to assess the following convergence properties. If the step size sequence γ_n is $(\ell^2 - \ell^1)$–summable, and the players' payoff estimates $\hat{u}^{(i)}(\alpha)$ are bounded and unbiased, the algorithm converges (a.s.) to a connected set of QRE of the game with rationality parameter $\theta = 1/T$. In particular, $p(n)$ converges within $\varepsilon(T)$ of a Nash equilibrium of the game and the error $\varepsilon(T)$ vanishes as $T \to 0$.

The previous algorithm can be adapted to the case where the only information at the players' disposal is the payoff of their chosen actions, possibly perturbed by some random noise process. It can also be adapted to incorporate asynchronous updates from the players as well as communication delays.

Further properties of this algorithm and its variants have been studied in Belmega and Mertikopoulos [2014]. They show that it minimizes regret in the framework MIMO network optimization.

REFERENCES

J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 27–46, 2008.

A. Antoniadis, E. Paparoditis, and T. Sapatinas. A functional wavelet–kernel approach for time series prediction. *Journal of the Royal Statistical Society: Series B*, 68:837–857, 2006.

A. Antoniadis, X. Brossat, J. Cugliari, and J. Poggi. Clustering functional data using wavelets. In *Proceedings of the Nineteenth International Conference on Computational Statistics (COMPSTAT)*, 2010.

A. Antoniadis, X. Brossat, J. Cugliari, and J. Poggi. Prévision d'un processus à valeurs fonctionnelles en présence de non stationnarités. Application à la consommation d'électricité. *Journal de la Société FranÃČÂğaise de Statistique*, 153:52–78, 2012.

A. Antoniadis, X. Brossat, J. Cugliari, and J. Poggi. Clustering functional data using wavelets. *International Journal of Wavelets, Multiresolution and Information Processing*, 11, 2013.

J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2635–2686, 2010.

J.-Y. Audibert, S. Bubeck, and G. Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39:31–45, 2014.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning Journal*, 47:235–256, 2002a.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32:48–77, 2002b.

K. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43:211–246, 2001.

M. Beckman, C. McGuire, and C. Winsten. *Studies in the Economics of Transportation.* Yale University Press, 1956.

V. Belmega and P. Mertikopoulos. Transmit without regrets: Online optimization in mimoâĂŞofdm cognitive radio systems. *IEEE Journal on Selected Areas in Communications*, 32, 2014.

M. Benaïm. Dynamics of stochastic approximation algorithms. *Séminaire de probabilités de Strasbourg*, 33, 1999.

M. Benaïm and M. Faure. Consistency of vanishingly smooth fictitious play. *Mathematics of Operations Research*, 38:437–450, 2013.

M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions. *I. SIAM Journal on Optimization and Control*, 44:328–348, 2005.

M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic Approximations and Differential Inclusions. Part II: Applications. *Mathematics of Operations Research*, 31:673–695, 2006.

A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007.

S Boucheron, G Lugosi, and P Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence.* Oxford University Press, 2013.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5:1–122, 2012.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

S. Bubeck, V. Perchet, and P. Rigollet. Bounded regret in stochastic multi-armed bandits. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pages 122–134, 2013.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, Cambridge, 2006.

N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

C.-K. Chiang, T Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 6.1–6.20, 2012.

H. Cho, Y. Goude, X. Brossat, and Q. Yao. Modeling and forecasting daily electricity load curves: a hybrid approach. *Journal of the American Statistical Association*, 108:7–21, 2013.

H. Cho, Y. Goude, X. Brossat, and Q. Yao. Modeling and forecasting daily electricity load using curve linear regression. In *Lecture Notes in Statistics: Modeling and Stochastic Learning for Forecasting in High Dimension*, 2014. To appear.

P. Coucheney, S. Durand, B. Gaujal, and C. Touati. General Revision Protocols in Best Response Algorithms for Potential Games. In IEEE Explore, editor, *Netwok Games, Control and OPtimization (NetGCoop)*, Trento, Italy, 2014a.

P. Coucheney, B. Gaujal, and P. Mertikopoulos. Penalty-regulated dynamics and robust learning procedures in games. Inria RR 33 pages, 3 figures, 2014b.

V. Dani, T. Hayes, and S. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 22*, pages 345–352, 2008.

M. Devaine, P. Gaillard, Y. Goude, and G. Stoltz. Forecasting electricity consumption by aggregating specialized experts. *Machine Learning*, 90:231–260, 2013.

D. P. Foster and R. V. Vohra. Regret in the on-line decision problem. *Games Econom. Behav.*, 29:7–35, 1999.

D. Fudenberg and D. M. Kreps. Learning mixed equilibria. *Games Econom. Behav.*, 5:320–367, 1993.

D. Fudenberg and D. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065 – 1089, 1995.

P. Gaillard and Y. Goude. Forecasting the electricity consumption by aggregating experts; how to design a good set of experts. 2014.

P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In *Proceedings of COLT*, 2014.

R.G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, 25:73–85, 1977.

S. Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. *Journal of Machine Learning Research*, 14:729–769, 2013.

J. Hannan. Approximation to Bayes risk in repeated play. In *Contributions to the Theory of Games*, volume 3 of *Annals of Mathematics Studies*, pages 97–139. Princeton University Press, Princeton, N. J., 1957.

S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68: 1127–1150, 2000.

S. Hart and A. Mas-Colell. Uncoupled dynamics cannot lead to nash equilibrium. *The American Economic Review*, 93:1830–1836, 2003.

S. Hart and A. Mas-Colell. Stochastic uncoupled dynamics and Nash equilibrium. *Games Econom. Behav.*, 57: 286–303, 2006.

E. Hazan. The convex optimization approach to regret minimization. *Optimization for machine learning*, page 287, 2012.

E. Hazan and S. Kale. Extracting certainty from uncertainty: regret bounded by variation in costs. *Machine Learning*, 80(2-3):165–188, 2010.

N. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems 18*, 2004.

J. Kwon and P. Mertikopoulos. A continuous-time approach to online optimization. *arXiv preprint arXiv:1401.6956*, 2014.

T. L. Lai and H. Robbins. Asymptotically optimal allocation of treatments in sequential experiments. In T. J. Santner and A. C. Tamhane, editors, *Design of Experiments: Ranking and Selection*, pages 127–142. 1984.

T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

G. Lugosi, S. Mannor, and G. Stoltz. Strategies for prediction under imperfect monitoring. *Math. Oper. Res.*, 33:513–528, 2008.

D. Monderer and L. Shapley. Potential games. *Games and economic behavior, Elsevier*, 14:124–143, 1996.

A. Orda, R. Rom, and N. Shimkin. Competitive routing in multeuser communication networks. *IEEE/ACM Trans. on Networking*, 1:510–521, 1993.

V. Perchet. No-regret with partial monitoring: Calibration-based optimal algorithms. *J. Mach. Learn. Res.*, 12:1893–1921, 2011.

V. Perchet. Exponential weight approachability, applications to calibration and regret minimization. *Dynamic Games And Applications*, 2014.

V. Perchet and P. Rigollet. The multi-armed bandit problem with covariates. *Ann. Statist.*, 41:693–721, 04 2013.

A. Pierrot and Y. Goude. Short-term electricity load forecasting with generalized additive models. In *Proceedings of ISAP power*, pages 593–600, 2011.

A. Pierrot, N. Laluque, and Y. Goude. Short-term electricity load forecasting with generalized additive models. In *Proceedings of the Third International Conference on Computational and Financial Econometrics (CFE)*, 2009.

R. Rosenthal. A class of games possessing pure-strategy nash equilibria. *Int. J. of Game Theory, Springer*, 2: 65–67, 1973.

T. Roughgarden. *Selfish Routing and the Price of Anarchy.* MIT Press, 2005.

A. Rustichini. Minimizing regret: the general case. *Games Econom. Behav.*, 29:224–243, 1999.

S. Sorin. Exponential weight algorithm in continuous time. *Mathematical Programming*, 116:513–528, 2009.

P. Taylor and L. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40: 145–156, 1978.

B. von Stengel. Computing equilibria for two-person games. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory*, pages 1723–1759. North-Holland, Amsterdam, 2002.

V. G. Vovk. Aggregating strategies. In *Proceedings of the Third Workshop on Computational Learning Theory*, pages 371–386, 1990.

J.G. Wardrop. Some theoretical aspects of road traffic research. part ii. *Proc. of the Institute of Civil Engineers*, 1:325–378, 1954.

O. Wintenberger. Optimal learning with bernstein online aggregation. Extended version available at arXiv:1404.1356 [stat.ML], 2014.

M. Woodroofe. A one-armed bandit problem with a concomitant variable. *J. Amer. Statist. Assoc.*, 74:799–806, 1979.